3D Reconstruction from Sketch with Hidden Lines by Two-Branch Diffusion Model

Yuta Fukushima[1] Anran Qi[1] I-Chao Shen[1] Yulia Gryaditskaya[2] Takeo Igarashi[1]

[1]The University of Tokyo
[2]CVSSP, Surrey Institute for People-Centred AI, University of Surrey, UK

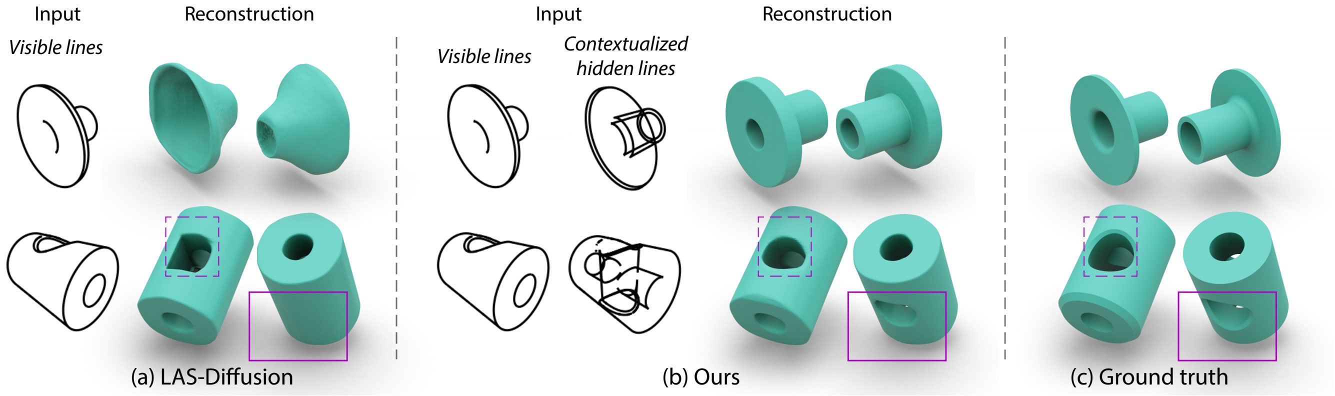


Figure 1: *Reconstruction results by (a) a LAS-Diffusion [ZPW*23] method from a sketch containing visible lines only and (b) our method from visible lines and contextualized hidden lines from a single viewpoint; (c) the ground truth. By explicitly considering hidden lines, our method can reconstruct the invisible structures of 3D objects from a single viewpoint, solving the inherent ambiguity of modelling occluded surfaces from a single view sketch. Observe the penetrating hole reconstructed using our method, highlighted by a box with a solid contour.*

Abstract

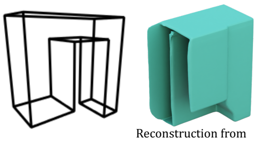
We present a method for sketch-based modelling of 3D man-made shapes that exploits not only the commonly considered visible surface lines but also the hidden lines typical for technical drawings. Hidden lines are used by artists and designers to communicate holistic shape structure. Given a single viewpoint sketch, leveraging such lines allows us to resolve the ambiguity of the shape's surfaces hidden from the observer. We assume that the separation into visible and hidden lines is given, and focus solely on how to leverage this information. Our strategy is to mingle two distinct diffusion networks: one generates denoized occupancy grid estimates from a visible line image, whilst the other generates occupancy grid estimates based on contextualized hidden lines unveiling the occluded shape structure. We iteratively merge noisy estimates from both models in a reverse diffusion process. Importantly, we demonstrate the importance of what we call a contextualized hidden lines image over just a hidden lines image. Our contextualized hidden lines image contains hidden lines and silhouette lines. Such contextualization allows us to achieve superior performance to a range of alternative configurations and reconstruct hidden holes and hidden surfaces.

CCS Concepts
• ***Computing methodologies*** → ***Artificial intelligence;*** • ***Computer graphics*** → *Shape modeling;*

1. Introduction

Sketching is often the first step in creating a 3D shape. This initial sketching phase provides a visual framework for ideas, allowing designers to explore and refine concepts rapidly. Subsequently, these preliminary sketches are transformed into detailed 3D models. In this work, we focus on how to leverage hidden lines typical to technical drawings for sketch-based 3D shape modelling. Hidden lines are used to convey information about surfaces that are not observed from the drawing viewpoint ([LGJ18] Chapter 6). Many of the existing sketch-based 3D shape modelling methods either leverage only visible surface lines [ZPW*23, PMKB23] or require multi-view inputs [LGK*17]. A few works [GHL*20, LPBM22] process sketches containing visible and hidden lines, but do not differentiate between them and assume that sketches are provided in a vector format with a known order of strokes. The raster sketch format remains the most widely adopted, and stroke order is challenging to infer for sketches drawn on paper. Therefore, we explore how to leverage hidden lines in raster sketches. Notably, none of the previous methods explicitly take advantage of hidden lines. Therefore, the first contribution of our work is to propose to explicitly utilize the hidden lines to infer the occluded details of 3D shapes.

We build our method on one of the state-of-the-art diffusion models for 3D shape generation from sketch inputs: LAS-Diffusion [ZPW*23]. We observe that LAS-Diffusion trained on all-line drawings struggles with understanding hidden lines in the all-line drawing, creating uncertainty on whether to establish a new surface based on these lines or regard them as part of the backside of an existing surface, ultimately leading to erroneous interpretations of the shape (see inset). To this end, our second contribution is that we propose to process visible and hidden lines separately, with two distinct branches that exchange information during the reverse diffusion process. Preliminary work on line type classification in bitmap sketches [HGB19] shows promising results, achieving 65% line type classification accuracy in their setting. Here we assume that classification is provided and leave automatic line classification for future work.



Reconstruction from LAS-Diffusion [ZPW*23]

Finally, we propose the usage of *contextualized hidden lines* and show the importance of contextualization. Namely, we show that when hidden lines are treated in isolation the network struggles to make a meaningful inference. However, when hidden lines are superimposed with silhouette lines our model efficiently combines information from visible and contextualized hidden lines.

Our results demonstrate that the incorporation of hidden lines enables a more precise reconstruction of the hidden structures of 3D objects (see Figures 1 and 3). We achieve the best Intersection over Union (IoU) and Light Field Descriptor (LFD) scores against various network configurations, improving over a single-branch baseline with a large margin.

2. Related Work

Works on sketch-based 3D shape modelling can be divided into two categories based on the sketch input format: vector [GHL*20, LPBM22] and bitmap [LGK*17, ZPW*23, PMKB23]. Vector format embeds the stroke order, which enables methods to process the strokes incrementally. Gryaditskaya *et al.* [GHL*20] lift strokes into 3D by computing their intended 3D intersections and depths. Li *et al.* [LPBM22] sequentially converts the strokes into CAD commands. However, stroke order is not accessible for sketches drawn on paper. Sketch-based modelling from a single bitmap sketch suffers from the inherent ambiguity of modelling occluded surfaces. Recent machine learning methods address this problem by either requiring multi-view inputs [LGK*17] or learning 3D shape class priors [ZPW*23]. However, man-made 3D shapes, especially those used in mechanical engineering, often contain complex geometric configurations that pose a significant challenge in learning meaningful 3D shape priors. We build our method around the observation that designers often sketch hidden lines in addition to visible lines to depict edges that are not visible from the view. This motivates us to investigate the use of hidden lines to infer the occluded details of 3D shapes that are not apparent from the visible lines alone.

3. Method

We build our model on the LAS-Diffusion model [ZPW*23]. Our method assumes that we know a separation into visible silhouette lines, other visible lines, and hidden lines (creases), that are provided to us as bitmap images or bitmap image annotations. We do not assume any information about individual strokes. Similarly to the LAS diffusion model, we leverage the information about the sketch viewpoint. However, it does not have to be precise and the user only needs to provide a rough guess of view information. The output is a 3D shape represented by a discrete signed distance field (SDF). A discrete signed distance function $g : z \in Z \mapsto \mathbb{R}$ is defined on a regular 3D grid Z, where $g(z)$ represents the signed distance from the centers of the grid cells to a closed manifold surface S.

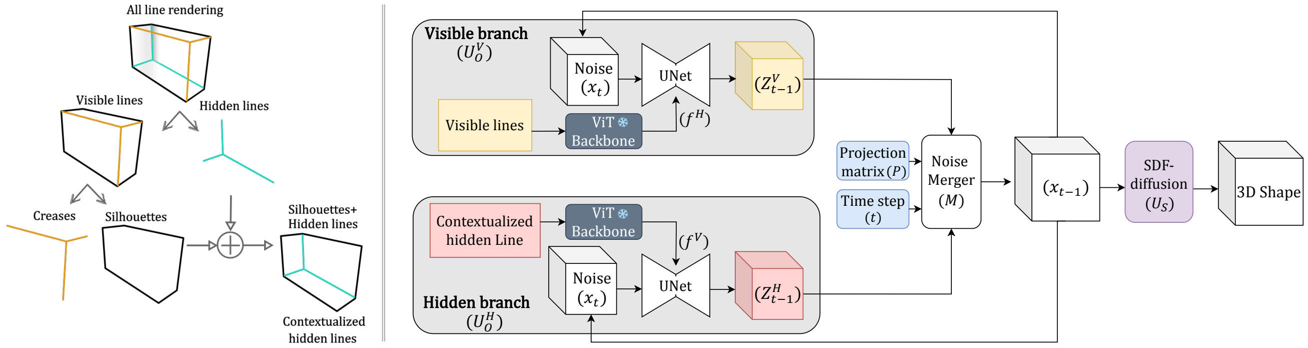
3.1. LAS-Diffusion Premilinaries

LAS-Diffusion model [ZPW*23] consists of two stages modelled with two networks: occupancy-diffusion network U_O, and SDF-diffusion network U_S. In the first stage, the model focuses on generating a low-resolution occupancy grid to approximate the overall shape structure. In the second stage, the network synthesizes a high-resolution signed distance field by refining occupied voxels from the first stage. The sketch conditioning information is taken into account only in the occupancy-diffusion network, U_O. In this work, our goal is to utilize hidden lines to infer the structural geometry of the 3D shape. Therefore, we inject hidden line information into the occupancy-diffusion network, which is the first stage of LAS-Diffusion.

3.2. Two-branch LAS-Diffusion Model

Figure 2(b) illustrates our network structure: It consists of two distinct occupancy-diffusion model branches, which we dub as *the visible branch* (U_O^V) and *the hidden branch* (U_O^H). The visible branch, U_O^V, takes a sketch of visible lines as input and predicts the occupancy grid of the 3D shape where only the information about the visible surface from a given viewpoint can be reconstructed reliably. The hidden branch, U_O^H, takes a sketch of contextualized hidden lines (silhouette and crease lines) as input. Its task is to estimate the geometry of surfaces complementary to the ones predicted with the visible branch, including holes and hidden surfaces.

The occupancy-diffusion network U_O in the LAS-Diffusion model [ZPW*23] for each time step t predicts a denoized result x_{t-1}. To perform conditional denoizing on input sketch, the occupancy-diffusion network is conditioned on view-projection matrix P and sketch patch features f extracted with the ViT (Vision Transformer) [DBK*21]: $x_{t-1} = U_O(x_t, f, t, P)$.



(a) Line types illustration

(b) Network structure illustration

Figure 2: *(a) We distinguish visible lines (creases and silhouettes) from hidden lines. We refer to the combination of hidden lines and silhouette lines as contextualized hidden lines. (b) Our model includes two parallel occupancy diffusion networks taking visible lines and contextualized hidden lines as input, respectively. A Noise Merger network (three Convolutional layers) merges the denoised predictions for a given time step. The output of this network is passed to the next stage SDF-diffusion network.*

In our two-branch architecture, each of the two branches for each time step t predicts denoized results z_{t-1}^V and z_{t-1}^H, respectively. We then introduce a noise merger network, M, consisting of three convolutional layers. It takes in z_{t-1}^V and z_{t-1}^H, time step t, and view-projection matrix P, and outputs a unified estimation x_{t-1}. In summary, one step of the reverse process of our two-branch LAS-Diffusion model can be expressed as follows:

$$z_{t-1}^V = U_O^V(x_t, f^V, t, P), \tag{1}$$

$$z_{t-1}^H = U_O^H(x_t, f^H, t, P), \tag{2}$$

$$x_{t-1} = M(z_{t-1}^V, z_{t-1}^H, t, P), \tag{3}$$

where f^V and f^H are image patch features extracted with the ViT [DBK*21] for the visible and hidden branches sketch inputs, respectively. Note that in the reverse process in Equations 1 and 2, we start from the unified noise volume x_t instead of z_t^V and z_t^H from the distinct occupancy-diffusion branches.

4. Experiments

4.1. Dataset

Due to the lack of datasets of technical drawings, we generate a dataset of synthetic line drawings using 3D shapes from the ABC dataset [KMJ*19], and NPR (Non-Photo Realistic) rendering. Since some of the models in the ABC dataset are extremely complex, their NPR renderings can get very cluttered. We select 3D models of limited complexity, by checking each shape against the following three criteria: (a) the number of parts is equal to one, (b) the number of surfaces is less than 10, and (c) the file size is smaller than 10 MB. In total, we selected 1,000 models from the ABC dataset [KMJ*19] for training and 300 models for testing. Some example shapes are shown in Figure 3. We render each shape from 50 different viewpoints, following Zheng *et al.* [ZPW*23]. We render silhouettes and visible creases as visible lines, and occluded creases as hidden lines (see Figure 2(a)) using Blender FreeStyle.

4.2. Implementation and Training Procedure

We use the same network configurations for the occupancy-diffusion model, U_O, and the SDF-diffusion model, U_S, following Zheng *et al.* [ZPW*23]. We refer the reader to their paper for more details. The noise merger, M, takes as input the concatenation of the two denoised occupancy grid results for a given time step from the occupancy-diffusion part, a projection matrix, and a time step. We implement M as three convolution layers with the number of neurons (32-8-1) and a kernel size of 3. The occupancy and SDF diffusion networks are initialized using the model pretrained on ShapeNet [CFG*15] dataset and the noise merger network is initialized using random weights. We trained the occupancy-diffusion branches U_O^V, U_O^H and the noise merger network M jointly for 300 epochs. Then, we further trained the SDF-diffusion network, U_S, for 300 epochs.

4.3. Comparisons

Alternative configurations We compare our method against the following three different configurations:

(a) Single-branch occupancy-diffusion (U_O), that takes in an NPR rendering of visible lines only;
(b) Single-branch occupancy-diffusion (U_O), that takes in an NPR rendering of visible lines and hidden lines as one image;
(c) Two-branch occupancy-diffusion (U_O^V and U_O^H), taking as input an NPR rendering of visible lines and an NPR rendering of hidden lines (silhouette lines are not included), respectively.

The outputs of (a)-(c) and of our method (Ours) are then refined by the SDF-Diffusion model, U_S, to get the discrete SDF values of the 3D shape, providing more details.

Quantitative comparison To measure the accuracy of the predicted 3D shapes, we use two quantitative evaluation metrics: Intersection-over-Union (IoU) and Light Field Distance (LFD) [CTSO03]. In Table 1, we show the IoU and LFD scores of the considered configurations. Our two-branch model using *contextualized hidden* lines (hidden + silhouette lines) outperforms all other configurations in both IoU and LFD scores. The two-branch architecture that takes in the second branch only hidden lines (c) has better IoU scores than single-branch configurations, but worse LFD scores. This shows the importance of *contextualizing* hidden lines in the second branch.

Proceedings published by Eurographics - The European Association for Computer Graphics.

	(a)	(b)	(c)	(Ours)
IoU ↑	0.607	0.613	0.651	**0.675**
LFD ↓	3201	3237	3272	**3123**

Table 1: *IoU and LFD scores of three alternative configurations (Section 4.3) and our method.*

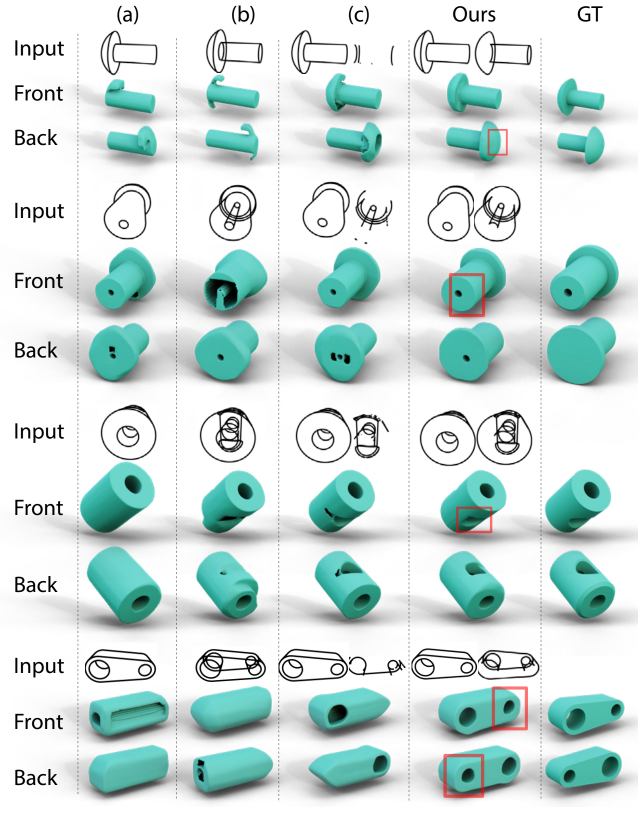
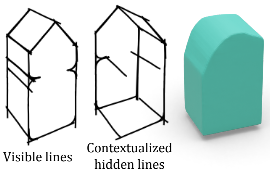


Figure 3: *Generated 3D shapes using different network configurations, described in Section 4.3. Red frames highlight the parts (holes and intricate structures) that can be more accurately reconstructed with our proposed method compared to other alternatives.*

Qualitative comparison Figure 3 shows qualitative results obtained with different configurations and demonstrates that our method can reconstruct holes and intricate structures more accurately than other configurations. Comparison of the results of (a) and other configurations, highlights the role of hidden lines in providing important geometric information about the occluded parts that visible lines cannot represent. However, as the results of (b) show, processing visible and hidden lines in the same image naively leads to erroneous reconstructions. Qualitative results further underline the advantage of our two-branch model and the usage of the proposed contextualized hidden lines.

4.4. Limitations and Future work

Our model is trained on synthetic data and might not generalize well to human-drawn sketches which usually contain highly distorted or over-sketched strokes. The inset shows a concept sketch example from [GSH*19], in which our model can capture the rough shape but fails to reconstruct the backside structure, possibly due to a large deviation in sketching style from the one in the training dataset. We plan to investigate this problem by using stylized sketches for training in the future. Lastly, our method can have difficulties reconstructing complex shapes, limited by the capacity of the diffusion model on sketch-based 3D shape generation techniques. However, we note that our main contribution is a novel solution for leveraging hidden lines to resolve the ambiguity of the shape's surfaces hidden from the observer. Further, as generation techniques progress rapidly, we hope that our work will propel the use of hidden lines in sketch-based modeling methods.



5. Acknowledgment

This work was partially supported by JST AdCORP, Grant Number JPMJKB2302, and JSPS Grant-in-Aid JP23K16921, Japan.

References

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 3

[CTSO03] Chen D.-Y., Tian X.-P., Shen Y.-T., Ouhyoung M.: On visual similarity based 3d model retrieval. In *CGF* (2003). 3

[DBK*21] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR* (2021). 2, 3

[GHL*20] Gryaditskaya Y., Hähnlein F., Liu C., Sheffer A., Bousseau A.: Lifting freehand concept sketches into 3d. *ACM TOG* (2020). 2

[GSH*19] Gryaditskaya Y., Sypesteyn M., Hoftijzer J. W., Pont S., Durand F., Bousseau A.: OpenSketch: A Richly-Annotated Dataset of Product Design Sketches. *ACM TOG* (2019). 4

[HGB19] Hähnlein F., Gryaditskaya Y., Bousseau A.: Bitmap or vector? A study on sketch representations for deep stroke segmentation. In *AFIG* (2019). 2

[KMJ*19] Koch S., Matveev A., Jiang Z., Williams F., Artemov A., Burnaev E., Alexa M., Zorin D., Panozzo D.: Abc: A big cad model dataset for geometric deep learning. In *CVPR* (2019). 3

[LGJ18] Lockhart S., Goodman M., Johnson C. M.: *Modern Graphics Communication*. Peachpit Press, 2018. 1

[LGK*17] Lun Z., Gadelha M., Kalogerakis E., Maji S., Wang R.: 3d shape reconstruction from sketches via multi-view convolutional networks. In *3DV* (2017). 2

[LPBM22] Li C., Pan H., Bousseau A., Mitra N. J.: Free2cad: Parsing freehand drawings into cad commands. *ACM TOG* (2022). 2

[PMKB23] Puhachov I., Martens C., Kry P. G., Bessmeltsev M.: Reconstruction of machine-made shapes from bitmap sketches. *ACM TOG* (2023). 2

[ZPW*23] Zheng X.-Y., Pan H., Wang P.-S., Tong X., Liu Y., Shum H.-Y.: Locally attentional sdf diffusion for controllable 3d shape generation. *ACM TOG* (2023). 1, 2, 3

Proceedings published by Eurographics - The European Association for Computer Graphics.